

Path Planning for Multiple Mobile Robots Using A* Algorithm

Z. BATIK GARIP^a, D. KARAYEL^{b,*}, S.S. OZKAN^b AND G. ATALI^c

^aSakarya University, Computer Programming, Sakarya, Turkey

^bSakarya University, Department of Mechatronics Engineering, Sakarya, Turkey

^cSakarya University, Electronic Automation, Sakarya, Turkey

One of the main problems in path planning for multiple mobile robots is to find the optimal path between two points so that robots can follow the shortest path and consume the least energy. Also, motion planning is important to avoid collisions if multiple mobile robots are running together within environments with obstacles. In this study, motion planning for multiple robots has been proposed. It is desired that robots could move in coordination with each other from the starting point to the destination point on a plane. The coordinates of the objects and mobile robots have been acquired using image processing with a single ceiling camera. The path planning for the shortest distance has been performed using A* algorithm in dynamic frame between robot-object and object-target point, respectively. A graphical user interface has been developed based on MATLAB GUI. It is hoped that the developed system will have a wide area of applications in industry and will make important contributions for the improvement of manufacturing, assembly, transportation and storage technologies.

DOI: [10.12693/APhysPolA.132.685](https://doi.org/10.12693/APhysPolA.132.685)

PACS/topics: 42.30.Tz, 45.40.Ln, 87.85.St, 87.55.de, 87.55.K-, 87.55.kd

1. Introduction

Today, the number of studies of robots has increased and robot application areas are rapidly expanding, as result of the technological development. It can be even said that robots have become a part of our daily lives. Robot studies focus on robot arm design and mobile robots. These studies continue with different control algorithms, while the first studies on this subject were mostly on the design and analysis of robot arms [1, 2].

Nowadays, mobile robots are being used in many sectors. Service robots, automated guided vehicles, unmanned bomb disposal robots and planet exploration robots are just some of the mobile robot applications. Path planning plays an important role in multiple mobile robot applications and there are many problems that must be overcome. Different methods have been used and new technologies have been developed according to application sectors. Genetic algorithm, potential field and heuristic solution are some of these methods.

Many research studies have been focused on this matter. Song et al. have developed a novel multimodal delayed particle swarm optimization algorithm, named MDPSO, for the global smooth path planning for mobile robots. This novel algorithm was developed in order to reduce the occurrence of local trapping phenomenon and to expand the search space in the searching process [3]. The free-space model, based on the Maklink graph theory was proposed for mobile robots in a study conducted by Zeng et al. Then, Dijkstra algorithm has been found, so that a collisionless path could be designed.

Finally, an appropriate path based on A* algorithm and genetic algorithm has been found using a hybrid approach [4]. Wu et al. have presented a kernel principal component analysis for robot localization with uncalibrated monocular visual information. The localization algorithm they developed in their work has performed better than standard SLAM algorithms [5]. Keysermann and Vargas have presented a novel architecture for associative learning and recall of different sensor and actuator patterns. The clustering process, developed by researchers, is performed by a topology-learning algorithm and the associative capabilities are implemented with a memory model [6].

A* algorithm and genetic algorithm, based on a hybrid method, was developed for optimal path planning for mobile robots by Zhang and et al. In their study, A* algorithm is utilized for fast search of a constrained shortest path, and the global optimal path is obtained by using genetic algorithm, to optimize the path [7]. Angular calculations with the distance calculations of the nearest nodes have been studied by Bhadoria and Singh to move the robot over the appropriate path, using a new heuristic function in addition to the traditional A* algorithm. It has been found that the proposed algorithm is 8% more effective compared to the traditional A* algorithm [8]. Zar and Sein have used A* algorithm to calculate optimal paths of bus routes in the city of Yangon. The desired route was introduced into a database and A* algorithm was used to calculate the optimal path [9].

This study is different from similar available studies, as shown below, and thus it is original. In this study, a path planning has been performed so that robots could move into coordinated manner on a plane from the starting

*corresponding author; e-mail: dkarayel@sakarya.edu.tr

point to target point. In the path planning, the coordinates of the mobile robot and the objects have been obtained by processing images of the movement area. For this process, only one camera, which was mounted on the upper part of the workspace, has been used. The shortest global path planning of multiple mobile robots has been performed considering the objects with the shortest distance and distance from objects to the target point in a dynamic environment using A* algorithm. Also, MATLAB based GUI (graphical user interface) has been prepared so that the developed method could be used easily and effectively.

2. Path planning system

The motion planning of mobile robots can be divided into two tasks, the positioning and the path planning. The stages of the development of path planning are shown in Fig. 1. The coordinates of robots and objects have been determined by positioning. Path planning process aims to find an appropriate path, so that each robot can go from the starting point to the subordinate target point (nearest object), and then it can proceed to the main target. In this manner, the motion planning can be described as path tracking in line with target. Path planning has been carried out in two steps. In the first step, the image of surrounding area has been obtained with a camera placed in the laboratory, as shown in Fig. 2.

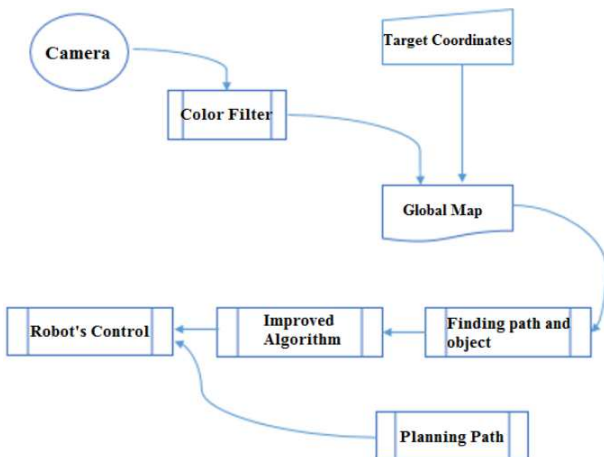


Fig. 1. Flow chart of path planning.

The obtained images have been entered as input data in the image processing function written using MATLAB, as shown in Fig. 3.

In the second step, the map has been included into the operation flow, for the path planning process, as shown in Fig. 1. In this study, A* algorithm has been applied for path planning of multiple mobile robots. It is aimed that the multiple mobile robots could find an optimal path and so they could go to their target by the shortest route. A* algorithm is described in detail in the following section.

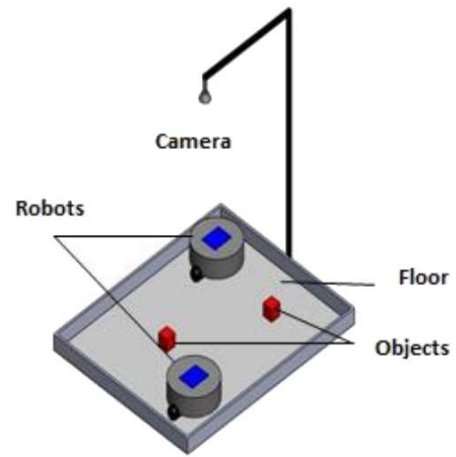


Fig. 2. The imaging apparatus installed in the laboratory.



Fig. 3. The block diagram of image processing function.

3. A* algorithm

Mobile robots generally consume more energy than the other robots during navigation. Therefore, efficient use of energy in mobile robots is important. Hence, it is preferred to move to the target position using the shortest route and the intelligent path planning algorithm should be developed. The path has been configured using the developed planning algorithm, based on heuristic approach, for the robot working in a real and dynamic environment. Finally, a least-cost path has to be found. Such algorithm was proposed by P. Hart, N. Nilsson and B. Raphael for the first time in 1968.

The shortest path is calculated as follows,

$$f(x) = g(x) + h(x),$$

where, $f(x)$ is the total cost function, $g(x)$ is distance function, and $h(x)$ is the heuristic function.

Distance function $g(x)$ describes the distance between the starting node and available current node, where the robot is located. The heuristic functions $h(x)$ represents the length of a straight line between the current node and the end node. This is also called the bird's-eye distance between the current node and the target. The algorithm proceeds from the first node to the node which has the lowest $f(x)$ value. A* algorithm flow chart is given in Fig. 4.

4. Robot operation system

The robot operating system (ROS) is a set of software libraries and tools that help to build robot applications. It is entirely open source, from drivers to state-of-the-art algorithms, and powerful developer tools. The ROS

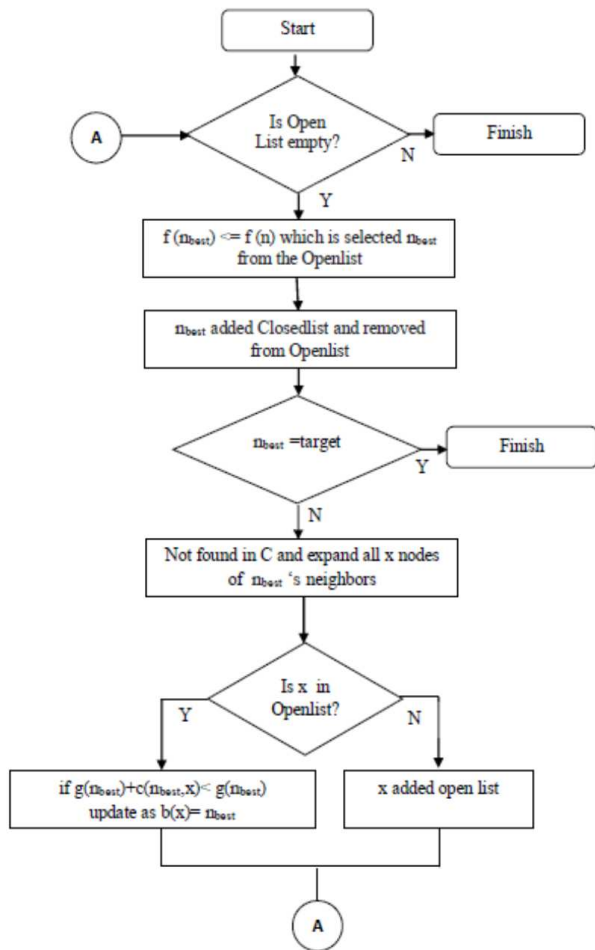


Fig. 4. Flow chart of A* algorithm.

is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms. ROS is easier to use, in comparison to general-purpose robot software. Therefore, ROS was built from the ground up, to encourage collaborative robotics software development. In other words, ROS is a collection of software frameworks for robot software development.

ROS provides standard operating system services such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. Running sets of ROS-based processes are represented in a graph architecture, where processing takes place in nodes that may receive, post and multiplex sensor, control, state, planning, actuator and other messages.

ROS contains many open source implementations of common robotics functionality and algorithms. Many packages are included as part of ROS distributions. Also, ROS tool allows users to plan complex trajectories on 3D surfaces and so it can be used for many purposes [10].

Meanwhile, MATLAB Robotics System Toolbox provides an interface between MATLAB and Simulink and

the ROS. Hence, it can be used to communicate with a ROS network. This feature allows to develop the robotics algorithms in MATLAB and Simulink [11].

5. Designing user interface

In this study, MATLAB based GUI was designed in order to use the system more easily. The user interface consists of three parts, such as global path planning, local path planning and camera image acquisition and processing. The images obtained by the image processing algorithm are shown in Fig. 5.

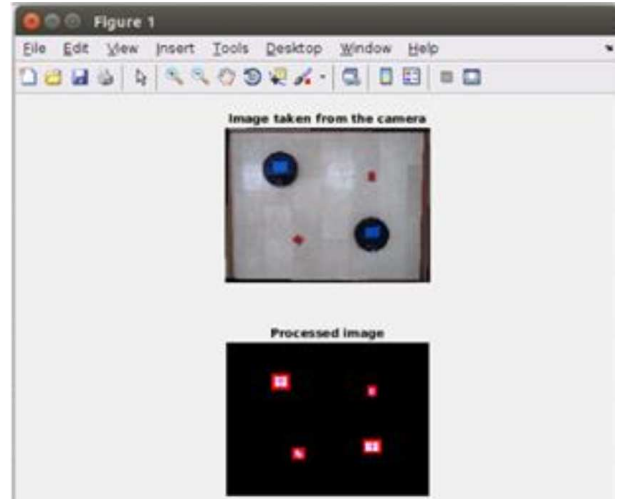


Fig. 5. Determination of position using image processing.

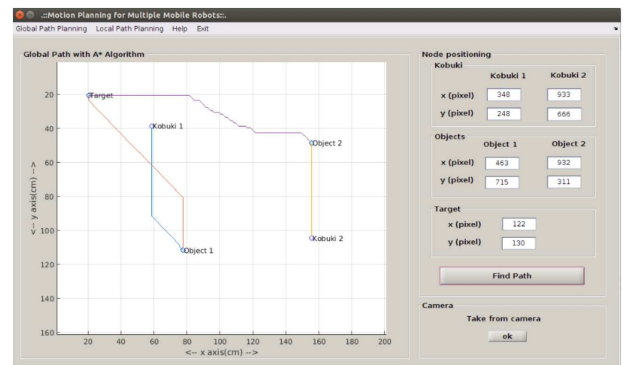


Fig. 6. Finding optimum path.

After the image processing stages, the x and y coordinates of nodes (Kobuki 1, Kobuki 2, Object 1, and Object 2) are obtained. Coordinates are converted into real-world values and so the positioning of objects is achieved. The optimum path is found after pressing the “Path Finder” button, using A* algorithm for Kobuki 1 and Kobuki 2, to the first stage and then to the common target points. The obtained routes are drawn on the working environment by using the developed system, as shown in Fig. 6.

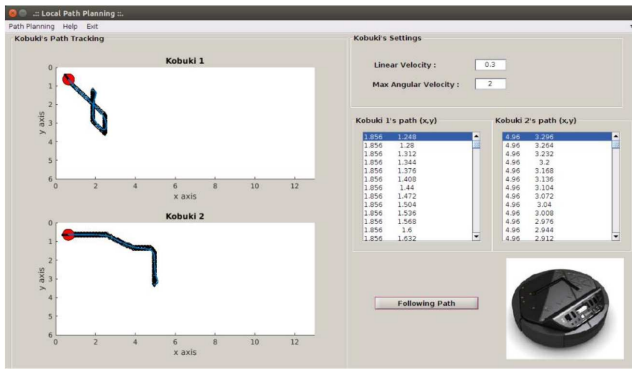


Fig. 7. The paths traveled by mobile robots.

When the “Follow path” button in the section of “Local path planning” is clicked, the path coordinates are obtained using A* algorithm and then the speed is calculated, using pure pursuit algorithm under laboratory conditions, and is sent to multiple mobile robots (Kobuki 1 and 2). The calculated linear and angular velocity values are sent to the mobile robots by means of ROS integrated with MATLAB. As shown in Fig. 7, the paths in the working area, traveled by mobile robots, and their coordinates (x, y) can be monitored by the user, via the developed system. Thus, the linear and angular velocities of the robots can be determined using the interface.

6. Conclusions

In this paper, a path planning approach based on A* algorithm for multiple mobile robots is proposed. It is shown that A* algorithm reduces the processing time considerably and thus A* algorithm can be used to find a path quickly. MATLAB based GUI has been designed and integrated with the system developed in this study.

Thus the system has become more efficient. It is hoped that the system will have a wide area of applications in many sectors and it will make important contribution for the improvement of transportation and storage technologies in industry.

References

- [1] N.G. Adar, R. Kozan, *Acta Phys. Pol. A* **130**, 269 (2016).
- [2] N.G. Adar, A. Egrisogut Tiryaki, R. Kozan, *Acta Phys. Pol. A* **128**, B-348 (2015).
- [3] Baoye Song, Zidong Wang, Lei Zou, *Cognitive Computat.* **9**, 5 (2017).
- [4] Cen Zeng, Qiang Zhang, Xiaopeng Wei, in: *3rd Int. Conf. Measuring Technology and Mechatronics Automation*, Vol. 3, IEEE, Shangshai 2011, p. 167.
- [5] Hua Wu, Yan-Xiong Wu, Chang-An Liu, Guo-Tian Yang, Shi-Yin Qin, *Cognitive Computat.* **8**, 856 (2016).
- [6] M.U. Keysermann, P.A. Vargas, *Cognitive Computat.* **7**, 414 (2015).
- [7] Liang Zhang, Huasong Min, Hongxing Wei, Haojun Huang, in: *IEEE Int. Conf. Robotics and Biomimetics*, IEEE, Guangzhou, China 2012.
- [8] A. Bhadoria, R.K. Singh, *Int. J. Intelligent Syst. Applicat. (IJISA)* **6**, 46 (2014).
- [9] Myat Thu Zar, Myint Sein, *Int. J. Adv. Computat. Eng. Networking* **3**, 7 (2015).
- [10] ROS (Robot Operation System), <http://www.ros.org>.
- [11] MATLAB, The MATHWORKS Inc., <http://www.mathworks.com>.